

SAFETY DEVICE FOR A STORED-PROGRAM CONTROL

Background Information

German Patent No. 44 06 094 describes a method and a device for real-time operation of a processor. In order to operate a processor in real time under a non-real time-capable operating system, the existing external hardware interrupt sources have 5 direct access to the non-maskable processor interrupt, bypassing the existing interrupt hardware and software.

Strict safety requirements of a stored-program control, however, call for additional measures in order to guarantee both the safety standard and the user-friendliness of a 10 stored-program control, which can be run on a conventional personal computer.

Summary Of The Invention

The safety device according to the present invention for a stored-program control has a controller which exchanges data with a stored-program control and, via a bus controller and a bus system, with the peripheral to be controlled. A memory is provided, in which 15 safety-relevant data of the stored-program control is stored, which the controller can access. In this preferably non-volatile memory, the data required for starting the stored-program control, such as remanent flags and important data modules, are stored. This additional storage allows the stored-program control to be quickly started in the event 20 of a malfunction of the personal computer assuming the functions of a stored-program control. A conventional personal computer can be easily retrofitted with the hardware configuration according to the present invention in the form of an appropriate plug-in card.

25 Another embodiment provides a monitor in addition to the controller, which exchanges data with a stored-program control and, via a bus system, with the peripheral to be controlled. It monitors a ready status signal generated by the stored-program control and transmitted to it by the controller. A contactor providing an output signal which

EL179956022US

displays the operability of the stored-program control is activated as a function of the monitor. For example, if the ready status signal is not detected during a predefined period in a manner that is recognized as valid, the contactor modifies its output signal, so that the user is informed that the stored-program control is in an irregular state. With
5 the help of the contactor output signal, the user can activate appropriate countermeasures or warning functions. The output signal is implemented via a zero potential contact.

According to one advantageous embodiment, the monitor activates a bus controller
10 controlling the bus system as a function of the ready status signal. If the ready status signal of the stored-program control allows the stored-program control to conclude that a malfunction has occurred in the stored-program control, the monitor activates the bus controller so that the latter sends an activation signal corresponding to a safety state to the peripheral.
15

Another embodiment provides an additional interface in addition to the controller which exchanges data with a stored-program control and, via a bus system, with the peripheral to be controlled. This interface receives at least one control signal, which is forwarded to the stored-program control via the controller. The user can access the
20 stored-program control, for example, via stop, on, or start commands, even if a non-real time-capable operating system, which normally ensures data exchange between user and stored-program control, is out of service.

In another embodiment, a real time controller is provided in addition to the controller
25 which exchanges data with a stored-program control and, via a bus system, with the peripheral to be controlled. The real time controller sends a control signal to a bus system of a personal computer, and the bus system of the personal computer allows data exchange between the controller and the stored-program control. This real time controller may form an embodiment as described under the related art. The control
30 signal ensures real time capability for a processor on which a non-real time capable and a real time capable operating system may be run. The two operating systems

cannot operate simultaneously without this real time controller.

The components described for the different embodiments can be integrated to form a single functional unit in any combination according to the user's desires.

5

According to one advantageous embodiment, the controller, the memory and the monitor are arranged on a single circuit board. This ensures a highly reliable embodiment.

10

According to another advantageous embodiment, the controller, the monitor, the contactor and the interface are integrated to form a functional unit on a circuit board. In addition to the safety-relevant function blocks, the interface and the contactor allow the user to access the monitor function and to issue direct commands to the stored-program control.

15

Brief Description Of The Drawings

Figure 1 shows a first block diagram of the device according to the present invention.

20

Figure 2 shows a second block diagram of the device according to the present invention.

Detailed Description

According to Figure 1, a controller 10 exchanges data with a stored-program control (not illustrated) via a computer bus system 12. Controller 10 ensures data exchange with a memory 14. Controller 10 exchanges data with a bus controller 18, which accesses a bus system 20, to which peripheral 32 to be controlled is connected. Interface 16 receives control signals 17, which are relayed to controller 10. Controller 10 forwards a wake-up signal 25 of the stored-program control to a monitor 24. As a function of the wake-up signal 25, monitor 24 activates bus controller 18. Another signal generated by monitor 24 is sent to both contactor 26 and an AND gate 28. Controller 10 provides AND gate 28 with a ready status signal 23 coming from the stored-

program control as another input signal. The output signal of AND gate 28 is sent to contactor 26, which generate an output signal 27, which can be picked up at two terminals. Controller 10, memory 14, interface 16, bus controller 18, real time controller 22, monitor 24, contactor 26 and AND gate 28 are arranged on a plug-in card 30.

5

J. 31
Figure 2 again shows the above-described components plug-in card 30, control signal 17, output signal 27, bus system 20 and peripheral 32 connected to the bus system. Plug-in card 30 is now inserted in the slot of a conventional computer 50 and connected to a display window 45 via the computer system 12 running under an operating system 43, to an SPS program 49, which may run under a real time operating system 47, a processor 51 and a main memory 53. A programming environment 41, not necessarily integrated in computer 50, also runs under operating system 43.
B.A

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

With the initialization of computer 50, operating system 43 and the respective display window 45 are started. Then real time operating system 47 is initialized, whereupon SPS program 49 is loaded from main memory 53 and bus system 20 is configured.

SPS program 49 starts at the point that depends on the status stored upon termination of the latest run. Program 49 is executed in real time and run cyclically until a terminate command appears. In this event, SPS program 49, the current data stored in the data modules and, as the case may be, the states in memory 14 are saved.

Additional safety-relevant functions, a display, and a user intervention option are implemented on plug-in card 30, which are elucidated below on the basis of Figure 1.

Controller 10 controls the data exchange between computer bus system 12 and bus system 20. The output states of peripheral 32 are cyclically read and forwarded to SPS program 49. SPS program 49 in turn delivers control signals for the inputs of peripheral 32. This data exchange is controlled by controller 10. In addition, controller 10 is responsible for the coupling with memory 14, which is preferably designed as a non-volatile, battery-buffered, remanent memory 14. Safety-relevant data, accessible to the SPS program, are stored in memory 14. These are preferably data that must be available even in the event of a power failure or a malfunction of computer 50 for quick restart, such as, for example, remanent flags or data modules in which characteristic peripheral data is permanently stored. Should computer 50 malfunction, the stored-program control can retrieve the data stored in memory 14 for the subsequent restart.

Control signals 17 are sent to controller 10 via interface 16. Control signals 17 can be influenced via switches. One input of interface 16, for example, can be used to set the stored-program control to stop mode. If the user activates the stop command via this control signal 17, controller 10 relays this signal to SPS program 49 via computer bus system 12, whereupon the SPS program jumps to the respective stop routine. The process is similar in the event of a second control signal 17, which is used as a start command to activate SPS program 49. This allows the user to influence the status of the stored-program control, bypassing programming environment 41.

Controller 10 relays wake-up signal 25 coming from the stored-program control to monitor 24. This is a cyclically occurring signal, a sync signal or a watchdog signal. Monitor 24 opens a time window of 2 seconds, for example, within which an edge of wake-up signal 25 must occur so that a conclusion can be made about the error-free 5 operation of the stored-program control. Otherwise, monitor 24 generates an emergency signal for bus controller 18, which activates an emergency function of bus controller 18. If bus controller 18 receives this emergency signal from monitor 24, it sets peripheral 32 to a predefined safe mode. This safe mode command is stored in bus controller 18.

10 Contactor 26 provides an output signal 27, which can be picked up by the user, preferably in the form of a zero-potential switching signal. Output signal 27 indicates whether the stored-program control is in an error-free state. The display of an error state of the stored-program control can be initiated via a signal directly generated by 15 monitor 24.

Contactor 26 provides an output signal 27, which can be picked up by the user, preferably in the form of a zero-potential switching signal. Output signal 27 indicates whether the stored-program control is in an error-free state. The display of an error state of the stored-program control can be initiated via a signal directly generated by 20 monitor 24. Controller 10 issues a software-generated error mode signal to monitor 24, which converts this signal via hardware into a control signal for contactor 26. As another alternative for generating an output signal 27 indicating the error state of the stored-program control is to supply the output signal of AND gate 28. To do so, a ready status signal 23 and the output signal of monitor 24 are AND-ed. The error state is then displayed by output signal 27 if the stored-program control issues no ready state signal 23 and also monitor 24 has recognized an error state. The appropriate display and warning functions can be activated via output signal 27.

25 Real time control 22 is also arranged on plug-in card 30, as described, for example, in German Patent No. 44 06 094. It issues a safety signal 21 to computer bus system 12, which ensures simultaneous operation of operating system 43 and real time operating system 47 without impairment of the real time capabilities of real time operating system 30 47.

Controller 10, memory 14, interface 16, bus controller 18, real time controller 22, monitor 24, contactor 26, and AND gate 28 are preferably arranged on a circuit board and thus form plug-in card 30.

- 5 The function of controller 10 and bus controller 18 can also be carried out by a single processor/module.